Multistep Direct Model Predictive Control for Power Electronics—Part 1: Algorithm

Tobias Geyer, *Senior Member, IEEE* and Daniel E. Quevedo, *Member, IEEE*

***Abstract*—For direct model predictive control with reference tracking of the converter current, we derive an efficient optimization algorithm that allows us to solve the control problem for very long prediction horizons. This is achieved by adapting sphere decoding principles to the underlying optimization problem. The proposed algorithm requires only few computations and directly provides the optimal switch positions. Our method is illustrated for the case of a variable speed drive system with a three-level voltage source converter. Since the computational burden of our algorithm is effectively independent of the number of converter output levels, the concept is particularly suitable for multi-level topologies with a large number of voltage levels.**

I. Introduction

During the past decade, model predictive control (MPC) for power electronics has received considerable attention; see, e.g., [1] and the references therein. MPC can be used for a large variety of topologies and in various operating conditions, its flexibility stemming from the online optimization of a suitable cost function. In particular, direct (or *finite control set*) MPC schemes tackle the current control and modulation problem in one computational stage and are, thus, promising. With direct MPC, the manipulated variable, chosen by the controller, is the inverter switch position, which is restricted to belong to a discrete and finite set [2]–[7]. A modulator is not needed.

A disadvantage of using MPC is that solving the underlying optimization problem and, thus, deriving the discrete manipulated variable, proves to be computationally challenging. Computational issues become especially important for long prediction horizons, since the number of possible switching sequences grows exponentially as the horizon length is increased [8]. As a result, when reference tracking of the converter currents is considered, the prediction horizon is usually set to one[1].

An alternative formulation of predictive control for power electronics and variable speed drives was presented in [11], [12]. In this approach, the machine's electromagnetic torque and stator flux magnitude, as well as the inverter's neutral point potential are kept within upper and lower bounds. Using the notion of extrapolation and restricting oneself to switching close to the bounds, large prediction horizons can be achieved [13], [14]. The same concept can be used to control the converter currents instead of the torque and stator flux [15]. Branch and bound methods can be added to tackle the high computational burden, which can typically be reduced by an order of magnitude [16]. This results in a family of MPC schemes with very long prediction horizons and a computational complexity that might be suitable for implementation on a modern DSP [17]. It has been shown that long prediction horizons lead to a significant performance improvement at steady-state operating conditions, lowering the current distortions and/or the switching frequency [18].

Despite the encouraging results in [18] for the scheme of [11], [12], in case of the basic direct MPC formulations in power electronics and drives (as used, e.g., in [2]–[7]), the question of whether longer horizons lead to performance improvements or not remains largely unanswered. We attribute the main reason for this knowledge gap to computational issues: In both MPC families, the optimization problem has traditionally been solved using some form of exhaustive search, i.e., the set of admissible switching sequences is enumerated, the corresponding response of the power electronic system is predicted, the cost function is evaluated and the switching sequence that yields the minimal cost is chosen as the optimal one. Enumeration is sometimes perceived as an "easy" task; this is a misconception since enumeration is applicable only to MPC problems featuring a limited number of switching sequences. Exhaustive enumeration is not practical for problems with thousands of sequences, which arise from MPC formulations with prediction horizons of four or more.

Motivated by the observations made above, this manuscript and its second part [19] examine the use of prediction horizons longer than one for direct MPC with reference tracking. To address computational issues, our work exploits the geometrical structure of the underlying optimization problem and presents an efficient optimization algorithm. The algorithm uses elements of sphere decoding [20] to provide optimal switching sequences, requiring only little computational resources. This enables the use of long prediction horizons in power electronics applications.

The proposed computational approach is derived for a linear system with a switched three-phase input with equal switching steps in all phases. Specifically, the present work focuses on a variable speed drive system, consisting of a three-level neutral point clamped voltage source inverter driving a medium-

T. Geyer is with ABB Corporate Research, Baden-Dättwil, Switzerland; e-mail: t.geyer@ieee.org

Daniel E. Quevedo is with the University of Newcastle, Australia; e-mail: daniel.quevedo@newcastle.edu.au

[1]The authors are aware of only two exceptions, namely [9], in which a horizon of $N = 2$ is used, and [10]. In the latter, a heuristic is used to reduce the number of switching sequences for longer horizons. Moreover, a two-step prediction approach has been proposed in [2]. In here, in a first step, the computation delay is compensated, followed by a standard predictive controller with $N = 1$. Therefore, this is considered to be an $N = 1$ approach.

978-1-4799-0336-8/13/$31.00 ©2013 IEEE

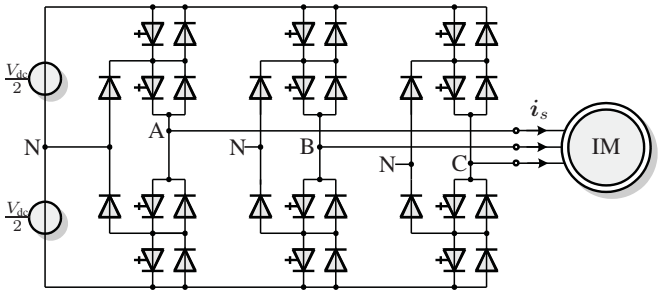


Fig. 1: Three-level three-phase neutral point clamped voltage source inverter driving an induction motor with a fixed neutral point potential

voltage induction machine. Our results in the second part [19] show that using prediction horizons larger than one does, in fact, provide significant performance benefits. In particular, at steady-state operation, the current distortions and/or the switching frequency can be reduced considerably with respect to direct MPC with horizon one. Indeed, in some cases, a steady-state performance can be achieved that is similar to the one of optimized pulse patterns.

Preliminaries

Throughout both papers, we will use normalized quantities. Extending this to the time scale t, one time unit corresponds to $1/\omega_b$ seconds, where ω_b is the base angular velocity. Additionally, we will use $\xi(t)$, $t \in \mathbb{R}$, to denote continuous-time variables, and $\xi(k)$, $k \in \mathbb{N}$, to denote discrete-time variables with the sampling interval T_s.

The $\alpha\beta$ Reference Frame: All variables $\boldsymbol{\xi}_{abc} = [\xi_a \ \xi_b \ \xi_c]^T$ in the three-phase system (abc) are transformed to $\boldsymbol{\xi}_{\alpha\beta} = [\xi_\alpha \ \xi_\beta]^T$ in the stationary orthogonal $\alpha\beta$ coordinates through $\boldsymbol{\xi}_{\alpha\beta} = \boldsymbol{P}\,\boldsymbol{\xi}_{abc}$, where

$$\boldsymbol{P} = \frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}. \tag{1}$$

Vector Norms: The 1-norm of a vector $\boldsymbol{\xi} \in \mathbb{R}^n$ is defined as $||\boldsymbol{\xi}||_1 \triangleq \sum_{i=1}^n |\xi_i|$, where ξ_i denotes the ith element of $\boldsymbol{\xi}$. The squared Euclidean norm of $\boldsymbol{\xi}$ is defined as $||\boldsymbol{\xi}||_2^2 \triangleq \boldsymbol{\xi}^T\boldsymbol{\xi} = \sum_{i=1}^n \xi_i^2$, and the squared norm weighted with the positive definite matrix $\boldsymbol{Q}$ is given by $||\boldsymbol{\xi}||_{\boldsymbol{Q}}^2 \triangleq \boldsymbol{\xi}^T\boldsymbol{Q}\boldsymbol{\xi}$. The infinity norm of $\boldsymbol{\xi}$ is defined as $||\boldsymbol{\xi}||_\infty \triangleq \max_i |\xi_i|$.

II. Drive System Case Study

Whilst the ideas of the present work can be applied to general ac-dc, dc-dc, dc-ac and ac-ac topologies with linear loads, including active front ends, inverters with RL loads and inverters with ac machines, we will focus our exposition on the setup described below.

A. Physical Model of the Inverter

As an illustrative example of a medium-voltage power electronic system, consider a variable speed drive consisting of a three-level neutral point clamped (NPC) voltage source inverter (VSI) driving an induction machine (IM), as depicted in Fig. 1. The total dc-link voltage $V_{\rm dc}$ is assumed to be constant and the neutral point potential N is fixed.

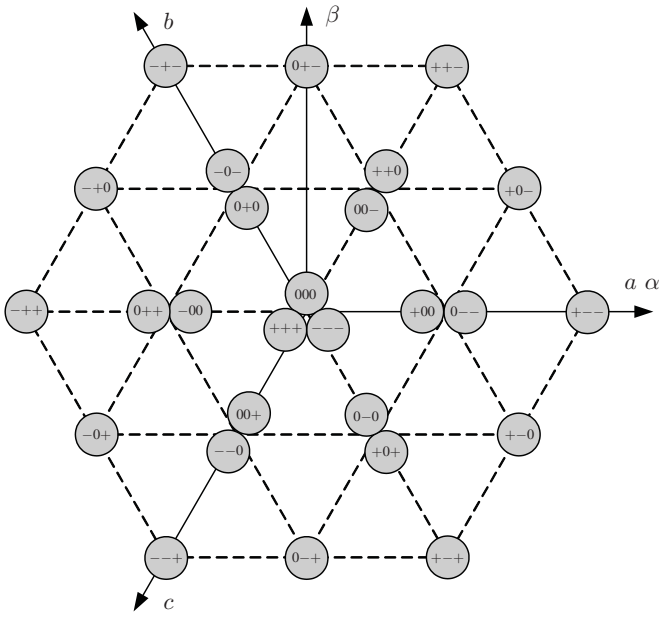


Fig. 2: Voltage vectors produced by a three-level inverter shown in the $\alpha\beta$ plane along with the corresponding values of the three-phase switch positions $\boldsymbol{u}$ (where '+' refers to '1' and '−' to '-1')

Let the integer variables u_a, u_b, $u_c \in \mathcal{U}$ denote the switch positions in the three phase legs, where for a three-level inverter the constraint set is given by

$$\mathcal{U} \triangleq \{-1, 0, 1\}. \tag{2}$$

In each phase, the values $-1, 0, 1$ correspond to the phase voltages $-\frac{V_{\rm dc}}{2}, 0, \frac{V_{\rm dc}}{2}$, respectively. Thus, the voltage applied to the machine terminals in orthogonal coordinates is

$$\boldsymbol{v}_{s,\alpha\beta} = \frac{1}{2}V_{\rm dc}\,\boldsymbol{u}_{\alpha\beta} = \frac{1}{2}V_{\rm dc}\,\boldsymbol{P}\,\boldsymbol{u} \tag{3}$$

with $\boldsymbol{u} \triangleq [u_a \ u_b \ u_c]^T$. The voltage vectors are shown in Fig. 2.

B. Physical Model of the Machine

The state-space model of a squirrel-cage induction machine in the stationary $\alpha\beta$ reference frame is summarized hereafter. For the current control problem at hand, it is convenient to choose the stator currents $i_{s\alpha}$ and $i_{s\beta}$ as state variables. The state vector is complemented by the rotor flux linkages $\psi_{r\alpha}$ and $\psi_{r\beta}$, and the rotor's angular velocity ω_r. The model input are the stator voltages $v_{s\alpha}$ and $v_{s\beta}$. The model parameters are the stator and rotor resistances R_s and R_r, the stator, rotor and mutual reactances X_{ls}, X_{lr} and X_m, respectively, the inertia J, and the mechanical load torque T_ℓ. All rotor quantities are referred to the stator circuit. In terms of the above quantities,

the continuous-time state equations are [21], [22]

$$\frac{\mathrm{d}\boldsymbol{i}_s}{\mathrm{d}t} = -\frac{1}{\tau_s}\boldsymbol{i}_s + \left(\frac{1}{\tau_r} - \omega_r \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\right)\frac{X_m}{D}\boldsymbol{\psi}_r + \frac{X_r}{D}\boldsymbol{v}_s \tag{4a}$$

$$\frac{\mathrm{d}\boldsymbol{\psi}_r}{\mathrm{d}t} = \frac{X_m}{\tau_r}\boldsymbol{i}_s - \frac{1}{\tau_r}\boldsymbol{\psi}_r + \omega_r \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\boldsymbol{\psi}_r \tag{4b}$$

$$\frac{\mathrm{d}\omega_r}{\mathrm{d}t} = \frac{1}{J}(T_e - T_\ell)\,, \tag{4c}$$

where we have used

$$X_s \triangleq X_{ls} + X_m,\ X_r \triangleq X_{lr} + X_m,\ D \triangleq X_s X_r - X_m^2. \tag{5}$$

(To simplify the notation, in (4) we dropped $\alpha\beta$ from the vectors $\boldsymbol{i}_s$, $\boldsymbol{\psi}_r$ and $\boldsymbol{v}_s$.) The transient stator time constant and the rotor time constant are equal to

$$\tau_s \triangleq \frac{X_r D}{R_s X_r^2 + R_r X_m^2} \quad \text{and} \quad \tau_r \triangleq \frac{X_r}{R_r}\,, \tag{6}$$

whereas the electromagnetic torque is given by

$$T_e = \frac{X_m}{X_r}(\psi_{r\alpha} i_{s\beta} - \psi_{r\beta} i_{s\alpha})\,. \tag{7}$$

III. Model Predictive Current Control

The control problem is formulated in the $\alpha\beta$ reference frame. Let $\boldsymbol{i}_s^*$ denote the reference of the instantaneous stator current, with $\boldsymbol{i}_s^* \triangleq [i_{s\alpha}^*\ \ i_{s\beta}^*]^T$. The objective of the current controller is to manipulate the three-phase switch position $\boldsymbol{u}$, such that the stator current $\boldsymbol{i}_s$ closely tracks its reference. At the same time, the switching effort, i.e., the switching frequency or the switching losses, are to be kept small. To avoid a shoot-through, direct switching between 1 and -1 in a phase leg is prohibited.

The block diagram of the model predictive current controller is shown in Fig. 3. As can be appreciated in that figure, the controller computes predicted trajectories of the variables of interest in order to optimize a performance criterion online. For the predictions, the measured stator current is required along with the rotor flux vector, which is typically obtained using a flux observer.

A. Prediction Model

The predictive current controller relies on an internal model of the physical drive system to predict future stator current trajectories. The rotor speed ω_r is assumed to be constant within the prediction horizon, which turns the speed into a time-varying parameter. The prediction horizon being less than one ms, this appears to be a mild assumption[2].

For our subsequent analysis, it is convenient to describe the system by introducing the following state vector of the drive model:

$$\boldsymbol{x} \triangleq [i_{s\alpha}\ i_{s\beta}\ \psi_{r\alpha}\ \psi_{r\beta}]^T. \tag{8}$$

[2]Nevertheless, including the speed as an additional state in the model might be necessary for highly dynamic drives and/or drives with a small inertia. The additional computational complexity this would entail is marginal.

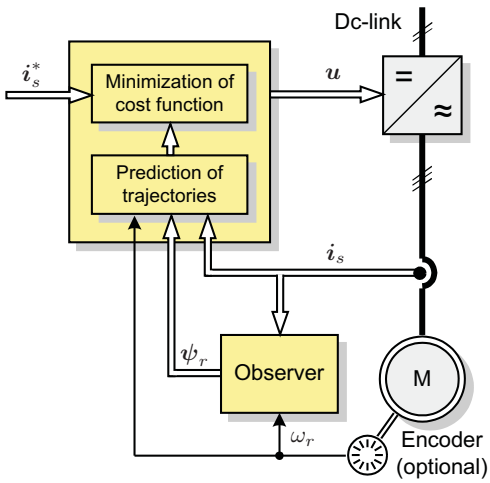


Fig. 3: Model predictive current control with reference tracking for the three-phase three-level NPC inverter with an induction machine

The stator current is taken as the system output vector, i.e., $\boldsymbol{y} = \boldsymbol{i}_s$, whereas the switch position $\boldsymbol{u}_{\alpha\beta}$ in the orthogonal coordinate system constitutes the input vector, which is provided by the controller.

Given the model described in Section II, in terms of $\boldsymbol{x}$, the continuous-time prediction model becomes

$$\frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t} = \boldsymbol{F}\,\boldsymbol{x}(t) + \boldsymbol{G}\,\boldsymbol{u}_{\alpha\beta}(t) \tag{9a}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}\,\boldsymbol{x}(t)\,, \tag{9b}$$

where the matrices $\boldsymbol{F}$, $\boldsymbol{G}$ and $\boldsymbol{C}$ are provided in the appendix[3].

By integrating (9a) from $t = kT_s$ to $t = (k+1)T_s$ and observing that during this time-interval $\boldsymbol{u}_{\alpha\beta}(t)$ is constant and equal to $\boldsymbol{u}_{\alpha\beta}(k)$, one obtains the discrete-time representation

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\,\boldsymbol{x}(k) + \boldsymbol{B}\,\boldsymbol{u}_{\alpha\beta}(k) \tag{10a}$$

$$\boldsymbol{y}(k) = \boldsymbol{C}\,\boldsymbol{x}(k) \tag{10b}$$

where $\boldsymbol{A} \triangleq \boldsymbol{e}^{\boldsymbol{F}T_s}$, $\boldsymbol{B} \triangleq -\boldsymbol{F}^{-1}\big(\boldsymbol{I} - \boldsymbol{A}\big)\boldsymbol{G}$[4] and $k \in \mathbb{N}$.

B. Cost function

The control problem at time-step k of tracking the current reference over a finite prediction horizon of length N can be addressed through minimization of the cost function

$$J = \sum_{\ell=k}^{k+N-1} ||\boldsymbol{i}_{e,abc}(\ell+1)||_2^2 + \lambda_u ||\boldsymbol{\Delta u}(\ell)||_2^2\,, \tag{11}$$

where the current error in abc-frame is defined as $\boldsymbol{i}_{e,abc} \triangleq \boldsymbol{i}_{s,abc}^* - \boldsymbol{i}_{s,abc}$, and the switching effort is defined as $\boldsymbol{\Delta u}(k) \triangleq$

[3]Note that $\boldsymbol{F}$ and $\boldsymbol{G}$ depend on the rotor speed ω_r and the dc-link voltage V_{dc}, respectively. Therefore, in a general setup, these two matrices need to be considered to be time-varying.

[4]Note that $\boldsymbol{e}$ refers to the matrix exponential, and $\boldsymbol{I}$ is the identity matrix of appropriate dimension (here 4×4). If matrix exponentials were to pose computational difficulties, the forward Euler approximation is usually sufficiently accurate for short sampling intervals of up to several tens of μs. In this case, the discrete-time system matrices are given by $\boldsymbol{A} \triangleq \boldsymbol{I} + \boldsymbol{F}T_s$ and $\boldsymbol{B} \triangleq \boldsymbol{G}T_s$.

$\boldsymbol{u}(k) - \boldsymbol{u}(k-1)$, thereby referring to the switch positions in the three phases a, b and c.[5] The first term in (11) penalizes the predicted three-phase current error at the time-steps $k+1$, $k+2$, $\ldots k+N$, using the squared Euclidean norm; the second term penalizes the switching effort at the time-steps k, $k+1$, $\ldots k+N-1$. The parameter $\lambda_u \geq 0$ is a tuning parameter, which adjusts the trade-off between the tracking accuracy (deviation of the current from its reference) and the switching effort.

Since in (8), the stator currents are represented in $\alpha\beta$ coordinates rather than in abc, it is convenient to express the first term in (11) in $\alpha\beta$ coordinates, too. For that purpose, we define

$$\boldsymbol{i}_{e,\alpha\beta} \triangleq \boldsymbol{i}^*_{s,\alpha\beta} - \boldsymbol{i}_{s,\alpha\beta}$$

and recall that $\boldsymbol{i}_{e,abc} = \boldsymbol{P}^{-1}\boldsymbol{i}_{e,\alpha\beta}$ with

$$\boldsymbol{P}^{-1} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}.$$

Noting that $\boldsymbol{P}^{-T}\boldsymbol{P}^{-1} = 1.5\boldsymbol{I}$, the first term in (11) can thus be rewritten as $||\boldsymbol{i}_{e,abc}||_2^2 = (\boldsymbol{i}_{e,abc})^T \boldsymbol{i}_{e,abc} = 1.5\,||\boldsymbol{i}_{e,\alpha\beta}||_2^2$. Omitting the factor 1.5 to simplify the expression, the equivalent cost function with the current error formulated in orthogonal coordinates becomes

$$J = \sum_{\ell=k}^{k+N-1} ||\boldsymbol{i}_{e,\alpha\beta}(\ell+1)||_2^2 + \lambda_u ||\boldsymbol{\Delta u}(\ell)||_2^2\,, \tag{12}$$

where

$$\boldsymbol{\Delta u}(\ell) = \boldsymbol{u}(\ell) - \boldsymbol{u}(\ell-1) \tag{13a}$$

$$\boldsymbol{i}_{e,\alpha\beta}(\ell+1) = \boldsymbol{i}^*_{s,\alpha\beta}(\ell+1) - \boldsymbol{C}\,\boldsymbol{x}(\ell+1) \tag{13b}$$

$$\boldsymbol{x}(\ell+1) = \boldsymbol{A}\,\boldsymbol{x}(\ell) + \boldsymbol{B}\,\boldsymbol{P}\,\boldsymbol{u}(\ell). \tag{13c}$$

C. Receding Horizon Optimization

We introduce the switching sequence

$$\boldsymbol{U}(k) = \begin{bmatrix} \boldsymbol{u}^T(k) & \ldots & \boldsymbol{u}^T(k+N-1) \end{bmatrix}^T,$$

which represents the sequence of inverter switch positions the controller has to decide upon. The optimization problem underlying direct MPC with current reference tracking can then be stated as

$$\boldsymbol{U}_{\text{opt}}(k) = \arg\min_{\boldsymbol{U}(k)} J \tag{14a}$$

$$\text{subj. to } \boldsymbol{U}(k) \in \mathbb{U} \tag{14b}$$

$$||\boldsymbol{\Delta u}(\ell)||_\infty \leq 1,\ \forall \ell = k, \ldots, k+N-1. \tag{14c}$$

The cost function J depends on the state vector $\boldsymbol{x}(k)$, the previously chosen switch position $\boldsymbol{u}(k-1)$ and the tentative switching sequence $\boldsymbol{U}(k)$. In (14b), $\mathbb{U} \triangleq \boldsymbol{\mathcal{U}} \times \cdots \times \boldsymbol{\mathcal{U}}$ is the N-times Cartesian product of the set $\boldsymbol{\mathcal{U}}$, where $\boldsymbol{\mathcal{U}}$ denotes the set of discrete three-phase switch positions. The latter is obtained from the single-phase constraints $\mathcal{U}$ via $\boldsymbol{\mathcal{U}} = \mathcal{U} \times \mathcal{U} \times \mathcal{U}$, as defined in (2). We refer to (14c) as switching constraints, which are imposed to avoid solutions leading to a shoot-through in the converter.

Following the receding horizon optimization principle, only the first element of the optimizing sequence $\boldsymbol{U}_{\text{opt}}(k)$ is applied to the semiconductor switches at time-step k; see, e.g. [1]. At the next time-step, $k+1$, and given new information on $\boldsymbol{x}(k+1)$, another optimization is performed, providing the optimal switch positions at time $k+1$. The optimization is repeated online and *ad infinitum*.

D. Obtaining the Switch Positions via Exhaustive Search

Due to the discrete nature of the decision variable $\boldsymbol{U}(k)$, the optimization problem (14) is difficult to solve, except for short horizons. In fact, as the prediction horizon is enlarged and the number of decision variables is increased, the (worst-case) computational complexity grows exponentially, thus, cannot be bounded by a polynomial, see also [8]. The difficulties associated with minimizing J become apparent when using exhaustive search. With this method, the set of admissible switching sequences $\boldsymbol{U}(k)$ is enumerated and the cost function evaluated for each such sequence. The switching sequence with the smallest cost is (by definition) the optimal one and its first element is chosen as the control input. At every time-step k, exhaustive search entails the following procedure:

1) Given the previously applied switch position $\boldsymbol{u}(k-1)$ and taking into account the constraints (14b) and (14c), determine the set of admissible switching sequences over the prediction horizon.
2) For each of these switching sequences, compute the state trajectory according to (13c) and the predicted evolution of the current error (13b).
3) For each switching sequence, compute the cost J according to (12).
4) Choose the switching sequence, $\boldsymbol{U}_{\text{opt}}(k)$, which minimizes the cost. Apply its first element, $\boldsymbol{u}_{\text{opt}}(k)$, to the converter.

At the next time-step, $k+1$, repeat the above procedure, using updated information on the current state vector, $\boldsymbol{x}(k+1)$, and reference trajectory, $\boldsymbol{i}^*_{s,\alpha\beta}(k+1), \ldots, \boldsymbol{i}^*_{s,\alpha\beta}(k+N+1)$.

It is easy to see that exhaustive search is computationally feasible only for very small horizons N, such as one or two. For $N = 5$, assuming a three-level converter and neglecting the switching constraint (14c), the number of switching sequences amounts to $1.4 \cdot 10^7$. This is clearly impractical, even when imposing (14c), which reduces the number of sequences by an order of magnitude.

Techniques from mathematical programming, such as branch and bound [16], [23], [24], can be used to reduce the computational burden of solving (14). In particular, off-the-shelf solvers such as CPLEX [25], include a wealth of smart heuristics and methods. However, none of the general methods take advantage of the particular structure of the

[5] Since in each phase, switching is only possible by one step up or down, i.e., we have $||\boldsymbol{\Delta u}(k)||_\infty \leq 1$, the 1-norm and the (squared) Euclidean norm of the switching effort yield the same cost: $||\boldsymbol{\Delta u}(k)||_1 = ||\boldsymbol{\Delta u}(k)||_2^2$.

optimization problem (14) and the fact that in MPC the solution is implemented in a receding horizon manner. One of the main aims of the present work is to shown how these distinguishing features of the problem at hand can be exploited in order to greatly reduce the computational burden, thereby enabling the use of large horizons in applications.

IV. Integer Quadratic Programming Formulation

In this section, we reformulate the optimization problem (14) in vector form and state it as a truncated integer least squares problem.

A. Optimization Problem in Vector Form

By successively using (13c), the state vector at time-step $\ell+1$ can be represented as a function of the state vector at time-step k and the switching sequence $\boldsymbol{U}(k)$ as follows:

$$\boldsymbol{x}(\ell+1) = \boldsymbol{A}\,\boldsymbol{x}(k) + \begin{bmatrix}\boldsymbol{A}^{\ell-k}\boldsymbol{B}\boldsymbol{P} & \dots & \boldsymbol{A}^{0}\boldsymbol{B}\boldsymbol{P}\end{bmatrix}\,\boldsymbol{U}(k) \quad (15)$$

with $\ell = k,\dots,k+N-1$. Let $\boldsymbol{Y}(k)$ denote the output sequence over the prediction horizon from time-step $k+1$ to $k+N$, i.e. $\boldsymbol{Y}(k) = [\boldsymbol{y}^T(k+1),\dots,\boldsymbol{y}^T(k+N)]^T$ and $\boldsymbol{Y}^*(k)$ correspondingly the output reference. Substituting (15) into (10b) gives

$$\boldsymbol{Y}(k) = \boldsymbol{\Gamma}\,\boldsymbol{x}(k) + \boldsymbol{\Upsilon}\,\boldsymbol{U}(k), \quad (16)$$

where the matrices $\boldsymbol{\Gamma}$ and $\boldsymbol{\Upsilon}$ are given in the appendix.

The dynamical evolution of the prediction model (13) can then be included in the cost function (12), yielding

$$J = ||\boldsymbol{\Gamma}\boldsymbol{x}(k) + \boldsymbol{\Upsilon}\boldsymbol{U}(k) - \boldsymbol{Y}^*(k)||_2^2 + \lambda_u||\boldsymbol{S}\boldsymbol{U}(k) - \boldsymbol{E}\boldsymbol{u}(k-1)||_2^2\,, \quad (17)$$

where $\boldsymbol{S}$ and $\boldsymbol{E}$ are defined in the appendix. As in (12), the first term in the cost function penalizes the predicted current tracking error, while the second term penalizes the switching effort.

The cost function (17) can be written in the compact form

$$J = \theta(k) + 2(\boldsymbol{\Theta}(k))^T\boldsymbol{U}(k) + ||\boldsymbol{U}(k)||_{\boldsymbol{Q}}^2 \quad (18)$$

with

$$\theta(k) \triangleq ||\boldsymbol{\Gamma}\boldsymbol{x}(k) - \boldsymbol{Y}^*(k)||_2^2 + \lambda_u||\boldsymbol{E}\boldsymbol{u}(k-1)||_2^2 \quad (19a)$$

$$\boldsymbol{\Theta}(k) \triangleq \big(\big(\boldsymbol{\Gamma}\boldsymbol{x}(k) - \boldsymbol{Y}^*(k)\big)^T\boldsymbol{\Upsilon} - \lambda_u\big(\boldsymbol{E}\boldsymbol{u}(k-1)\big)^T\boldsymbol{S}\big)^T \quad (19b)$$

$$\boldsymbol{Q} \triangleq \boldsymbol{\Upsilon}^T\boldsymbol{\Upsilon} + \lambda_u\boldsymbol{S}^T\boldsymbol{S}\,. \quad (19c)$$

Completing the squares shows that

$$J = (\boldsymbol{U}(k) + \boldsymbol{Q}^{-1}\boldsymbol{\Theta}(k))^T\boldsymbol{Q}(\boldsymbol{U}(k) + \boldsymbol{Q}^{-1}\boldsymbol{\Theta}(k)) + \text{const}(k)\,. \quad (20)$$

Note that the constant term in (20) is time-varying; it is a function of $\boldsymbol{x}(k)$ and $\boldsymbol{u}(k-1)$, but independent of $\boldsymbol{U}(k)$.

B. Solution in Terms of the Unconstrained Optimum

The *unconstrained* optimum of (14) is obtained by minimization, *neglecting* the constraints (14b) and (14c), thus allowing $\boldsymbol{U}(k) \in \mathbb{R}^3 \times \cdots \times \mathbb{R}^3$. Since $\boldsymbol{Q}$ is positive definite, it follows directly from (20) that the unconstrained solution at time-step k is given by

$$\boldsymbol{U}_{\text{unc}}(k) = -\boldsymbol{Q}^{-1}\boldsymbol{\Theta}(k)\,. \quad (21)$$

Since the first element of the unconstrained switching sequence $\boldsymbol{U}_{\text{unc}}(k)$ does not meet the constraints (14b) and (14c), it cannot be directly used as gating signals to the semiconductor switches, but $\boldsymbol{U}_{\text{unc}}(k)$ can be used to state the solution to the *constrained* optimization problem (14)—including the constraints (14b) and (14c)—as shown next.

Following the derivation as in [8], [26], the cost function (20) can be rewritten by inserting (21) as follows:

$$J = (\boldsymbol{U}(k) - \boldsymbol{U}_{\text{unc}}(k))^T\boldsymbol{Q}(\boldsymbol{U}(k) - \boldsymbol{U}_{\text{unc}}(k)) + \text{const}(k)\,. \quad (22)$$

Since $\boldsymbol{Q}$ is (by definition) symmetric and positive definite for $\lambda_u > 0$, there exists a unique *invertible* and *lower triangular* matrix $\boldsymbol{H} \in \mathbb{R}^{3N\times 3N}$, which satisfies:

$$\boldsymbol{H}^T\boldsymbol{H} = \boldsymbol{Q}. \quad (23)$$

The matrix $\boldsymbol{H}$ can be calculated by noting that its inverse, $\boldsymbol{H}^{-1}$, is also lower triangular and is provided by the following Cholesky decomposition of $\boldsymbol{Q}^{-1}$ [27]:

$$\boldsymbol{H}^{-1}\boldsymbol{H}^{-T} = \boldsymbol{Q}^{-1}. \quad (24)$$

In terms of $\boldsymbol{H}$ and

$$\bar{\boldsymbol{U}}_{\text{unc}}(k) \triangleq \boldsymbol{H}\boldsymbol{U}_{\text{unc}}(k), \quad (25)$$

the cost in (22) can be rewritten as

$$J = (\boldsymbol{H}\boldsymbol{U}(k) - \bar{\boldsymbol{U}}_{\text{unc}}(k))^T(\boldsymbol{H}\boldsymbol{U}(k) - \bar{\boldsymbol{U}}_{\text{unc}}(k)) + \text{const}(k)\,. \quad (26)$$

and the optimization problem (14) amounts to finding

$$\boldsymbol{U}_{\text{opt}}(k) = \arg\min_{\boldsymbol{U}(k)} ||\boldsymbol{H}\boldsymbol{U}(k) - \bar{\boldsymbol{U}}_{\text{unc}}(k)||_2^2 \quad (27)$$

subject to (14b) and (14c). Thus, we have rewritten the optimization problem as a (truncated) *integer least-squares* problem. Interestingly, various efficient solution algorithms for (27) subject to (14b)—but not taking into account (14c)—have been developed in recent years; see, e.g., [**?**], [20] and references therein. In Section V, we will tailor one such algorithm to the optimization problem of interest.

V. An Efficient Method for Calculating the Optimal Switch Positions

In this section, we show how to adapt the sphere decoding algorithm [20], [28] to find the optimal switching sequence $\boldsymbol{U}_{\text{opt}}(k)$. The algorithm is based on branch and bound techniques and is—as will be illustrated in the second part of this paper [19]—by far more efficient than the explicit enumeration method described in Section III-D. For ease of notation, throughout this section, we will write $\boldsymbol{U}$ instead of $\boldsymbol{U}(k)$.

A. Preliminaries and Key Properties

The basic idea of the algorithm is to iteratively consider candidate sequences, say $\boldsymbol{U} \in \mathbb{U}$, which belong to a sphere of radius $\rho(k) > 0$ centered in $\bar{\boldsymbol{U}}_{\text{unc}}(k)$,

$$\|\boldsymbol{H}\boldsymbol{U} - \bar{\boldsymbol{U}}_{\text{unc}}(k)\|_2 \leq \rho(k)\,, \tag{28}$$

and which satisfy the switching constraint (14c).

A key property used in sphere decoding is that, since $\boldsymbol{H}$ is triangular, for a given radius, identifying candidate sequences which satisfy (28) is very simple. In our case, $\boldsymbol{H}$ is lower triangular, and (28) can be rewritten as

$$\rho^2(k) \geq (\bar{U}_1 - H_{(1,1)}U_1)^2 + (\bar{U}_2 - H_{(2,1)}U_1 - H_{(2,2)}U_2)^2 + \dots \tag{29}$$

where $\bar{U}_i$ denotes the i-th element of $\bar{\boldsymbol{U}}_{\text{unc}}(k)$, U_i is the i-th element of $\boldsymbol{U}$, and $H_{(i,j)}$ refers to the (i,j)-th entry of $\boldsymbol{H}$. Therefore, the solution set of (28) can be found by proceeding in a sequential manner somewhat akin to Gaussian elimination, in the sense that at each step only a one-dimension problem needs to be solved; for details, see [20].

To determine $\boldsymbol{U}$, the algorithm requires an initial value for the radius used at time k. On the one hand, the radius $\rho(k)$ should be as small as possible, enabling us to remove as many candidate solutions *a priori* as possible. On the other hand, $\rho(k)$ must not be too small, to ensure that the solution set is non-empty. We propose to choose the initial radius by using the following *educated guess* for the optimal solution

$$\boldsymbol{U}_{\text{sub}}(k) = \begin{bmatrix} \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \mathbf{0} & \boldsymbol{I} \\ \mathbf{0} & \dots & \dots & \mathbf{0} & \boldsymbol{I} \end{bmatrix} \boldsymbol{U}_{\text{opt}}(k-1), \tag{30}$$

which is obtained by shifting the previous solution by one time-step and repeating the last switch position. This is in accordance with the receding horizon paradigm used in MPC. Since the optimal solution at the previous time-step satisfies both constraints (14b) and (14c), the shifted guess automatically meets these constraints, too. Thus, $\boldsymbol{U}_{\text{sub}}(k)$ is a feasible solution candidate of (27). Given (30), the initial value of $\rho(k)$ is then set to

$$\rho(k) = \|\boldsymbol{H}\,\boldsymbol{U}_{\text{sub}}(k) - \bar{\boldsymbol{U}}_{\text{unc}}(k)\|_2\,. \tag{31}$$

B. Modified Sphere Decoding Algorithm

At each time-step k, the controller first uses the current state $\boldsymbol{x}(k)$, the future reference values $\boldsymbol{Y}^*(k)$, the previous switch position $\boldsymbol{u}(k-1)$ and the previous optimizer $\boldsymbol{U}_{\text{opt}}(k-1)$ to calculate $\boldsymbol{U}_{\text{sub}}(k)$, $\rho(k)$ and $\bar{\boldsymbol{U}}_{\text{unc}}(k)$; see (30), (31), (25), (21), and (19b). The optimal switching sequence $\boldsymbol{U}_{\text{opt}}(k)$ is then obtained by invoking the following algorithm:

$$\boldsymbol{U}_{\text{opt}}(k) = \text{MSPHDEC}(\emptyset, 0, 1, \rho^2(k), \bar{\boldsymbol{U}}_{\text{unc}}(k)), \tag{32}$$

where $\emptyset$ is the empty set[6].

[6]The notation $\boldsymbol{H}_{(i,1:i)}$ refers to the first i entries of the i-th row of $\boldsymbol{H}$; similarly, $\boldsymbol{U}_{1:i}$ are the first i elements of the vector $\boldsymbol{U}$.

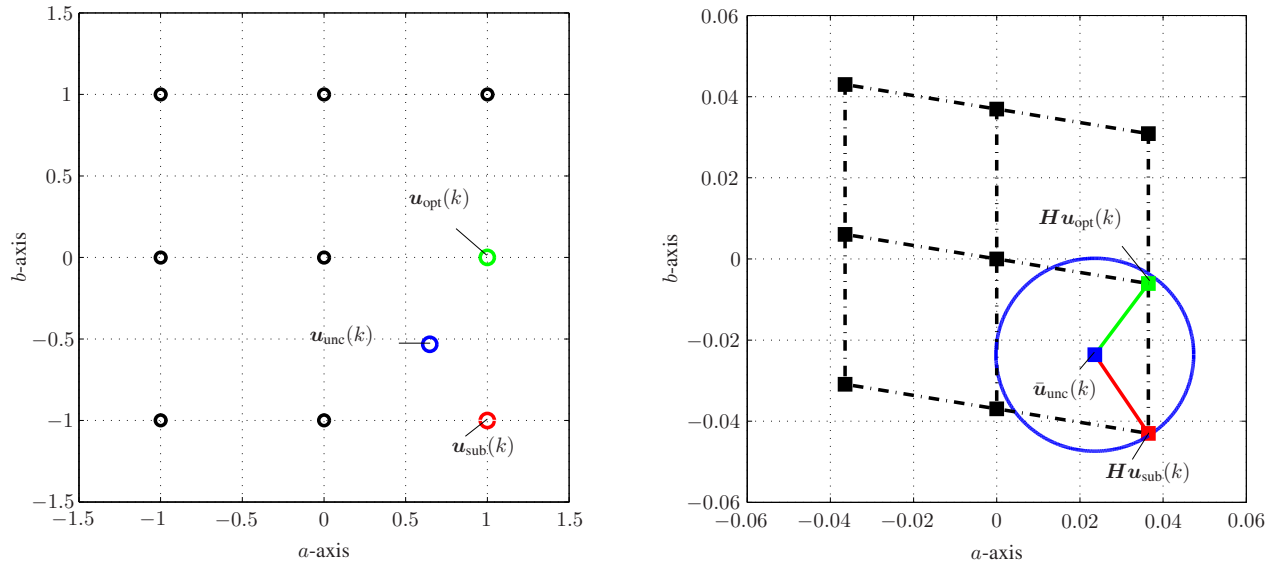
function $\boldsymbol{U}_{\text{OPT}}$ = MSPHDEC($\boldsymbol{U}$, d^2, i, ρ^2, $\bar{\boldsymbol{U}}_{\text{unc}}$)
  for each $u \in \mathcal{U}$ **do**
    $U_i = u$
    $d'^2 = ||\bar{U}_i - \boldsymbol{H}_{(i,1:i)}\boldsymbol{U}_{1:i}||_2^2 + d^2$
    if $d'^2 \leq \rho^2$ **then**
      if $i < 3N$ **then**
        MSPHDEC($\boldsymbol{U}, d'^2, i+1, \rho^2, \bar{\boldsymbol{U}}_{\text{unc}}$)
      else
        if $\boldsymbol{U}$ meets (14c) **then**
          $\boldsymbol{U}_{\text{OPT}} = \boldsymbol{U}$
          $\rho^2 = d'^2$
        end if
      end if
    end if
  end for
end function

As can be seen in this algorithm, the proposed modification to sphere decoding operates in a recursive manner. Starting with the first component, the switching sequence $\boldsymbol{U}$ is built component by component, by considering the admissible single-phase switch positions in the constraint set $\mathcal{U}$. If the associated squared distance is smaller than the current value of ρ^2, then we proceed to the next component. Once the last component, i.e., U_{3N}, has been reached, meaning that $\boldsymbol{U}$ is of full dimension $3N$, then $\boldsymbol{U}$ is a candidate solution. If $\boldsymbol{U}$ meets the switching constraint (14c) and if the distance is smaller than the current optimum, then we update the incumbent optimal solution $\boldsymbol{U}_{\text{opt}}$ and also the radius ρ.

The computational advantages of this algorithm stem from adopting the notion of branch and bound [23], [24]. Branching is done over the set of single-phase switch positions $\mathcal{U}$; bounding is achieved by considering solutions only within the sphere of current radius, see (28). If the distance d' exceeds the radius, a certificate has been found that the branch (and all its associated switching sequences) provides only suboptimal solutions, i.e., solutions that are worse than the incumbent optimum. Therefore, without having explored this branch, it can be pruned and removed from further consideration. During the optimization procedure, whenever a better incumbent solution is found, the radius is reduced and the sphere thus tightened, so that the set of candidate sequences is as small as possible, but non-empty. The majority of the computational burden relates to the computation of d' via evaluation of the terms $\boldsymbol{H}_{(i,1:i)}\boldsymbol{U}_{1:i}$. Thanks to (29), d' can be computed sequentially, by computing only the squared addition due to the ith component of $\boldsymbol{U}$. In particular, the sum of squares in d, accumulated over the layers 1 to $i-1$, does not need to be recomputed.

C. Illustrative Example for the Horizon $N = 1$ Case

To provide additional insight in the operation of the algorithm, we give an illustrative example of one problem instance. Consider the horizon $N = 1$ case with the sampling interval $T_s = 25\,\mu$s and the penalty $\lambda_u = 1 \cdot 10^{-3}$. Assuming a drive



(a) Optimization problem in the orthogonal coordinate system

(b) Optimization problem in the transformed coordinate system created by $\boldsymbol{H}$

Fig. 4: Visualization of the sphere decoding algorithm in the ab-plane for the horizon $N = 1$ case

system with a three-level inverter as in Fig. 1, the set of single-phase switch positions is $\mathcal{U} = \{-1, 0, 1\}$. We use the same drive parameters as in [19].

The set of admissible three-phase switch positions $\boldsymbol{u}(k) \in \boldsymbol{\mathcal{U}}$ is shown in Fig. 4(a) as black circles. To simplify the exposition, only the ab-plane is shown in this figure, neglecting the c-axis. Suppose that $\boldsymbol{u}_{\text{opt}}(k-1) = [1\ 0\ 1]^T$ and that the problem instance at time-step k yields the unconstrained solution $\boldsymbol{u}_{\text{unc}}(k) = [0.647\ -0.533\ -0.114]^T$, shown as a blue circle in the figure. Rounding $\boldsymbol{u}_{\text{unc}}(k)$ to the next integer values leads to the possible feasible solution $\boldsymbol{u}_{\text{sub}}(k) = [1\ -1\ 0]^T$, which corresponds to the red circle. It turns out, however, that the optimal solution is $\boldsymbol{u}_{\text{opt}}(k) = [1\ 0\ 0]^T$, indicated by the green circle.

The sphere decoding problem is solved in the transformed coordinate system, which is created by the generator matrix

$$\boldsymbol{H} = \begin{bmatrix} 36.45 & 0 & 0 \\ -6.068 & 36.95 & 0 \\ -5.265 & -5.265 & 37.32 \end{bmatrix} \cdot 10^{-3},$$

see (23). Using $\boldsymbol{H}$, the integer solutions $\boldsymbol{u} \in \boldsymbol{\mathcal{U}}$ in the orthogonal coordinate system can be transformed to $\boldsymbol{H}\boldsymbol{u}$, which are shown as black squares in Fig. 4(b) and connected by the dash-dotted lines. The coordinate system created by $\boldsymbol{H}$ is slightly skewed, but almost orthogonal, with the angle between the axes being $98.2°$ for the chosen parameters. It can be shown that increasing λ_u results in this angle converging towards $90°$.

The optimal solution $\boldsymbol{u}_{\text{opt}}(k)$ is obtained by minimizing the distance between the unconstrained solution and the sequence of integer switch positions in the transformed coordinate system. The initial value of $\rho(k)$ results from (31) and is equal to 0.638. This defines a ball of radius $\rho(k)$ around $\bar{\boldsymbol{u}}_{\text{unc}}(k) = \boldsymbol{H}\boldsymbol{u}_{\text{unc}}(k)$, which is shown in the ab-plane in Fig. 4(b) as the blue circle. This ball reduces the set of possible solutions from $3^3 = 27$ elements to two, since only two transformed integer solutions $\boldsymbol{H}\boldsymbol{u}(k)$ lie within the sphere, namely $\boldsymbol{H}\boldsymbol{u}_{\text{opt}}(k)$ (the green square) and $\boldsymbol{H}\boldsymbol{u}_{\text{sub}}(k)$ (the red square). The algorithm sequentially computes the distances between $\bar{\boldsymbol{u}}_{\text{unc}}(k)$ and each of these two points. These distances are indicated by the green and red line, respectively. The green line is slightly shorter than the red one. Therefore, minimizing the distance yields the optimal solution $\boldsymbol{u}_{\text{opt}}(k) = [1\ 0\ 0]^T$ and not the (suboptimal) naïvely rounded switch position $\boldsymbol{u}_{\text{sub}}(k) = [1\ -1\ 0]^T$.

VI. Conclusions

This manuscript addresses the major, and so far unsolved, problem of efficiently solving the optimization problem of direct model predictive control schemes with very long prediction horizons. As was shown, this can be achieved by adopting the notion of sphere decoding and by tailoring it to the power electronics problem at hand. Sphere decoding is effectively a smart branch and bound method. It is expected that sphere decoding will enable the use of long prediction horizons in power electronics, by facilitating the solution of the optimization problem within one sampling interval of, say, $25\ \mu$s.

The method proposed and results obtained in this paper are directly applicable to both the machine-side inverter in an ac drive setting, as well as to grid-side converters. The concepts can also be used for converter topologies other than the neutral point clamped converter—indeed, they are particularly promising for topologies with a high number of voltage levels.

As shown in the second part of this paper [19], long prediction horizons improve the converter performance at steady-state operating conditions, by either reducing the switching frequency or the total harmonic distortion (THD) of the current, or both. As a result, direct MPC can outperform space vector modulation and carrier-based PWM. In some cases, the performance of direct MPC may even approach the one of optimized pulse patterns, which are generally considered to provide the upper bound on the attainable steady-state performance.

ACKNOWLEDGMENT

This research was supported under Australian Research Council's Discovery Projects funding scheme (project number DP110103074). The first author gratefully acknowledges a research grant from ABB Corporate Research, Switzerland, while being with the University of Auckland, New Zealand.

VII. APPENDIX

The matrices corresponding to the continuous-time prediction model (9) are

$$\boldsymbol{F} = \begin{bmatrix} -\frac{1}{\tau_s} & 0 & \frac{X_m}{\tau_r D} & \omega_r \frac{X_m}{D} \\ 0 & -\frac{1}{\tau_s} & -\omega_r \frac{X_m}{D} & \frac{X_m}{\tau_r D} \\ \frac{X_m}{\tau_r} & 0 & -\frac{1}{\tau_r} & -\omega_r \\ 0 & \frac{X_m}{\tau_r} & \omega_r & -\frac{1}{\tau_r} \end{bmatrix}, \tag{33a}$$

$$\boldsymbol{G} = \frac{X_r}{D}\frac{V_{\mathrm{dc}}}{2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \ \boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{33b}$$

The matrices used in (16) and (17) are the following:

$$\boldsymbol{\Upsilon} = \begin{bmatrix} \boldsymbol{CBP} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{CABP} & \boldsymbol{CBP} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & & \vdots \\ \boldsymbol{CA}^{N-1}\boldsymbol{BP} & \boldsymbol{CA}^{N-2}\boldsymbol{BP} & \cdots & \boldsymbol{CBP} \end{bmatrix}$$

$$\boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{CA} \\ \boldsymbol{CA}^2 \\ \vdots \\ \boldsymbol{CA}^N \end{bmatrix}, \boldsymbol{S} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ -\boldsymbol{I} & \boldsymbol{I} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{I} \end{bmatrix}, \ \boldsymbol{E} = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \vdots \\ \boldsymbol{0} \end{bmatrix}.$$

REFERENCES

[1] P. Cortés, M. P. Kazmierowski, R. M. Kennel, D. E. Quevedo, and J. Rodríguez. Predictive control in power electronics and drives. *IEEE Trans. Ind. Electron.*, 55(12):4312–4324, Dec. 2008.

[2] S. Kouro, P. Cortés, R. Vargas, U. Ammann, and J. Rodríguez. Model predictive control—a simple and powerful method to control power converters. *IEEE Trans. Ind. Electron.*, 56(6):1826–1838, Jun. 2009.

[3] P. Lezana, R. P. Aguilera, and D. E. Quevedo. Model predictive control of an asymmetric flying capacitor converter. *IEEE Trans. Ind. Electron.*, 56(6):1839–1846, Jun. 2009.

[4] R. P. Aguilera and D. E. Quevedo. On stability and performance of finite control set MPC for power converters. In *Workshop on Pred. Control of Electr. Drives and Power Electron.*, pages 55–62, Munich, Germany, Oct. 2011.

[5] P. Cortés, J. Rodríguez, D. E. Quevedo, and C. Silva. Predictive current control strategy with imposed load current spectrum. *IEEE Trans. Power Electron.*, 23(2):612–618, Mar. 2008.

[6] D. E. Quevedo, R. P. Aguilera, M. A. Pérez, P. Cortés, and R. Lizana. Model predictive control of an AFE rectifier with dynamic references. *IEEE Trans. Power Electron.*, 27(7):3128–3136, Jul. 2012.

[7] J. Rodríguez and P. Cortés. *Predictive control of power converters and electrical drives*. Intersci. Publ. John Wiley & Sons Inc., 2012.

[8] D. E. Quevedo, G. C. Goodwin, and J. A. De Doná. Finite constraint set receding horizon quadratic control. *Int. J. Robust Nonlin. Contr.*, 14(4):355–377, Mar. 2004.

[9] P. Cortés, J. Rodríguez, S. Vazquez, and L.G. Franquelo. Predictive control of a three-phase UPS inverter using two steps prediction horizon. In *Proc. IEEE Int. Conf. Ind. Technol.*, pages 1283–1288, Viña del Mar, Chile, Mar. 2010.

[10] P. Stolze, P. Landsmann, R. Kennel, and T. Mouton. Finite-set model predictive control of a flying capacitor converter with heuristic voltage vector preselection. In *Proc. IEEE Int. Conf. on Power Electron. and ECCE Asia*, Jun. 2011.

[11] T. Geyer. *Low Complexity Model Predictive Control in Power Electronics and Power Systems*. PhD thesis, Autom. Control Lab. ETH Zurich, 2005.

[12] G. Papafotiou, T. Geyer, and M. Morari. A hybrid model predictive control approach to the direct torque control problem of induction motors (invited paper). *Int. J. of Robust Nonlinear Control*, 17(17):1572–1589, Nov. 2007.

[13] T. Geyer, G. Papafotiou, and M. Morari. Model predictive direct torque control—Part I: Concept, algorithm and analysis. *IEEE Trans. Ind. Electron.*, 56(6):1894–1905, Jun. 2009.

[14] T. Geyer. Generalized model predictive direct torque control: Long prediction horizons and minimization of switching losses. In *Proc. IEEE Conf. Decis. Control*, pages 6799–6804, Shanghai, China, Dec. 2009.

[15] T. Geyer. Model predictive direct current control: Formulation of the stator current bounds and the concept of the switching horizon. *IEEE Ind. Appl. Mag.*, 18(2):47–59, Mar./Apr. 2012.

[16] T. Geyer. Computationally efficient model predictive direct torque control. *IEEE Trans. Power Electron.*, 26(10):2804–2816, Oct. 2011.

[17] G. Papafotiou, J. Kley, K. G. Papadopoulos, P. Bohren, and M. Morari. Model predictive direct torque control—Part II: Implementation and experimental evaluation. *IEEE Trans. Ind. Electron.*, 56(6):1906–1915, Jun. 2009.

[18] T. Geyer. A comparison of control and modulation schemes for medium-voltage drives: Emerging predictive control concepts versus PWM-based schemes. *IEEE Trans. Ind. Appl.*, 47(3):1380–1389, May/Jun. 2011.

[19] T. Geyer and D.E. Quevedo. Multistep direct model predictive control for power electronics—Part 2: Analysis. In *Proc. IEEE Energy Convers. Congr. Expo.*, Denver, CO, USA, Sep. 2013.

[20] B. Hassibi and H. Vikalo. On the sphere-decoding algorithm I. Expected complexity. *IEEE Trans. Sign. Process.*, 53(8):2806–2818, Aug. 2005.

[21] J. Holtz. The representation of AC machine dynamics by complex signal graphs. *IEEE Trans. Ind. Electron.*, 42(3):263–271, Jun. 1995.

[22] P. C. Krause, O. Wasynczuk, and S. D. Sudhoff. *Analysis of Electric Machinery and Drive Systems*. Intersci. Publ. John Wiley & Sons Inc., 2nd edition, 2002.

[23] E. L. Lawler and D. E. Wood. Branch and bound methods: A survey. *Op. Res.*, 14(4):699–719, Jul./Aug. 1966.

[24] L. G. Mitten. Branch–and–bound methods: General formulation and properties. *Op. Res.*, 18(1):24–34, Jan./Feb. 1970.

[25] IBM ILOG, Inc. *CPLEX 12.5 User Manual*. Somers, NY, USA, 2012.

[26] G.C. Goodwin, D.Q. Mayne, K.-Y. Chen, C. Coates, G. Mirzaeva, and D.E. Quevedo. An introduction to the control of switching electronic systems. *Ann. Rev. in Control*, 34(2):209–220, Dec. 2010.

[27] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, UK, 1985.

[28] U. Fincke and M. Pohst. Improved methods for calculating vectors of short length in a lattice, including a complexity analysis. *Math. Comput.*, 44(170):463–471, Apr. 1985.